¿Cuál es el objetivo principal de la aplicación?

El objetivo principal de una API de Texto a Voz (TTS, por sus siglas en inglés) es convertir texto escrito en voz sintetizada de manera automática. Esta API permiten a los desarrolladores integrar capacidades de síntesis de voz en aplicaciones, sitios web, dispositivos y otros sistemas.

¿Cuál es el problema o necesidad que busca resolver?

Crear facilidad a los usuarios para una rápida aplicación de textos escritos para ser convertidos en audios estilos podcasts.

¿Quiénes son los usuarios de la aplicación?

Personas interesadas en aprender y conocer todos los recursos aplicativos por la aplicación.

¿Cuáles son sus características y necesidades específicas?

Accesibilidad

Interfaz de usuario hablada

Contenido multimedia

Localización y personalización

Automatización de tareas

¿Qué funcionalidades se requieren para cumplir con el objetivo de la aplicación?

Síntesis de voz

Selección de voz y configuración

Calidad de voz y naturalidad

Personalización de la voz

Integración y facilidad de uso

Escalabilidad y rendimiento

¿Cómo debe ser la interfaz de usuario?

Esta interfaz trata de ser simple para que los usuarios puedan trabajar de manera sencilla.

¿Qué elementos gráficos y de diseño son importantes para los usuarios?

Los textos sean comprensibles y entendibles para una mejor interacción en la API.

¿Hay alguna tecnología o herramienta específica que deba ser utilizada en el desarrollo de la aplicación?

Hasta el momento solo se lleva la base de la aplicación por consiguiente aun no se me hace fácil responder esa pregunta.

Backlog de producto:

Síntesis de voz: Desarrollar la capacidad de convertir texto escrito en voz sintetizada de manera automática.

Selección de voz y configuración: Permitir a los usuarios elegir entre diferentes voces y configurar opciones relacionadas con la síntesis de voz.

Calidad de voz y naturalidad: Mejorar la calidad y naturalidad de la voz sintetizada para lograr una experiencia más realista.

Personalización de la voz: Proporcionar opciones de personalización para adaptar la voz sintetizada a las preferencias del usuario.

Integración y facilidad de uso: Facilitar la integración de la API en diferentes aplicaciones y sistemas, y asegurar que sea fácil de utilizar.

Escalabilidad y rendimiento: Optimizar la capacidad de la API para manejar un alto volumen de solicitudes y garantizar un rendimiento eficiente.

Backlog de sprint:

Sprint 1:

1. Configurar el entorno de desarrollo.
2. Establecer la estructura de la base de la aplicación.
3. Implementar la funcionalidad básica de síntesis de voz.

Sprint 2:

1. Realizar pruebas de calidad y naturalidad de la voz sintetizada.
2. Agregar opciones de selección de voz y configuración.
3. Mejorar la integración y la facilidad de uso.

Sprint 3:

1. Incorporar funciones de personalización de la voz.
2. Optimizar el rendimiento y la escalabilidad de la API.
3. Realizar pruebas de aceptación y correcciones de errores.

Sprint 4:

1. Preparar la documentación del producto.
2. Realizar pruebas de integración.
3. Realizar pruebas de rendimiento y escalabilidad adicionales.

Sprint 5:

1. Finalizar las correcciones de errores identificados.
2. Realizar pruebas de aceptación finales.
3. Preparar el lanzamiento de la versión inicial del producto.

Es importante destacar que la duración de cada sprint puede variar dependiendo de los recursos disponibles y de la complejidad de las tareas. Además, es posible que se requieran sprints adicionales o que se realicen ajustes en función de los resultados y requisitos específicos del proyecto.

Esquema de Usuario:

const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({

username: { type: String, required: true },

password: { type: String, required: true },

email: { type: String, required: true },

createdAt: { type: Date, default: Date.now },

});

const User = mongoose.model('User', userSchema);

module.exports = User;

Esquema de Configuración de Voz:

const mongoose = require('mongoose');

const voiceConfigSchema = new mongoose.Schema({

userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

voiceName: { type: String, required: true },

language: { type: String, required: true },

speed: { type: Number, default: 1 },

volume: { type: Number, default: 1 },

});

const VoiceConfig = mongoose.model('VoiceConfig', voiceConfigSchema);

module.exports = VoiceConfig;



